**第2章 矩阵代数**

**1，求解逆矩阵**

**1.1，行列式**

方阵A的行列式通常表示为detA。方阵A是可逆的，当且仅当detA≠0。

矩阵的行列式有一种递归定义。4x4矩阵的行列式要根据3x3矩阵的行列式来定义。

余子阵：nxn的矩阵A，余子阵ij即为从A中去除第i行和第j列的(n-1)x(n-1)矩阵。

detA = A1j(-1)1+jdet1j

**1.2，余子式矩阵**

元素Aij的代数余子式：Cij=(-1)i+jdetij称为元素Aij的代数余子式。

如果为矩阵A中的每个元素分别计算出Cij，并将它置于CA中第i行，第j列的相应位置，那么获得矩阵A的代数余子式矩阵。

**1.3，伴随矩阵**

若取矩阵CA的转置矩阵，将得到矩阵A的伴随矩阵，记作A\*=CAT。

**1.4，逆矩阵**

**A-1=A\*/detA**

**2，用DirectXMath库处理矩阵**

**2.1 矩阵类型**

XMMATRIX类来表示4x4矩阵。

XMMATRIX由4个XMVECTOR实例所构成，并借此来使用SIMD技术。

可以用构造函数或XMMatrixSet函数来创建XMMATRIX实例。

建议使XMFLOAT4X4来存储类中的矩阵类型数据成员。

XMFLOAT4X4和XMMATRIX转换的函数：

inline XMMATRIX XM\_CALLCONV XMLoadFloat4x4(const XMFLOAT4X4\* pSource)

inline void XM\_CALLCONV XMStoreFloat4x4(XMFLOAT4X4\* pDestination, FXMMATRIX M)

**2.2 实用函数**

DirectXMath库包含了下列与矩阵相关的实用函数：

inline XMMATRIX XM\_CALLCONV XMMatrixIdentity()

inline bool XM\_CALLCONV XMMatrixIsInfinite(FXMMATRIX M)

inline XMMATRIX XM\_CALLCONV XMMatrixMultiply(FXMMATRIX M1, CXMMATRIX M2)

inline XMMATRIX XM\_CALLCONV XMMatrixTranspose(FXMMATRIX M)

//返回(det M,det M,det M,det M)

inline XMVECTOR XM\_CALLCONV XMMatrixDeterminant(FXMMATRIX M)

inline XMMATRIX XM\_CALLCONV XMMatrixInverse(XMVECTOR\* pDeterminant, FXMMATRIX M) //返回逆矩阵

**2.3 XMMATRIX参数**

假设传入函数的FXMVECTOR参数不超过两个，则第一个XMMATRIX参数应当为FXMMATRIX,其余XMMATRIX参数均应为CXMMATRIX。

//在32位的Windows系统上，\_\_fastcall调用约定通过寄存器传递前3个XMVECTOR参数，其余的参数则存在堆栈上

typedef const XMMATRIX& FXMMATRIX;

typedef const XMMATRIX& CXMMATRIX;

//在32位的Windows系统上，\_\_vectorcall调用约定通过寄存器传递前6个XMVECTOR参数，其余的参数则存在堆栈上

typedef const XMMATRIX FXMMATRIX;

typedef const XMMATRIX& CXMMATRIX;

DirectXMath建议用户总是在构造函数中采用CXMMATRIX类型来获取XMMATRIX参数，而且对于构造函数也不需要使用XM\_CALLCONV约定注解。